THAT WHICH IS CLAIMED:

1. A system for managing the display of information output by a computer program on a display terminal comprising:

data modules stored on a computer-readable medium containing text and graphical data used by the computer program to display information on the display terminal; and

a display management module stored on a computer-readable medium in communication with the computer program, display terminal, and data modules, wherein said display management module, based on commands output by the computer program, accesses the data modules and displays the data associated with the commands on the display terminal thereby eliminating the requirement that the text and graphical data for displaying information on the display terminal be contained in the computer program.

2. A system according to Claim 1, wherein at least one data module is a language data module including data strings representing language data, wherein each data string is stored in said language data module and designated by a token, and wherein to display a data string, said display management module receives a token associated with the data string from the computer program and a command to display the data string and based on the token accesses said language data modules, retrieves the data string associated with the token, and displays the data string on the display terminal.

3. A system according to Claim 2 further comprising:

a plurality of language data modules associated with said display management module, wherein each language data module includes data strings representing language data in a selected language; and

a main language module header stored on a computer-readable medium and associated with said display management module comprising individual pointers indicating the location in a computer-readable storage medium in which each language data module is located, wherein to display a data string in a selected language, said display management module receives a token from the computer program and a command to display the data string, wherein said display management module accesses

-40-

said main language module header and retrieves the pointer associated with the language data module corresponding to a preselected desired language for displaying the data string, and wherein said display management module using the pointer, accesses the language data module, retrieves the data string associated with the token, and displays the data string on the display terminal in the desired language.

4. A system according to Claim 1, further comprising a font module stored in a computer-readable medium and associated with said display management module, wherein said font module contains font data for displaying 256 standard and extended ASCII characters.

5. A system according to Claim 4, wherein at least one data module is a language data module comprising:

a string data area stored on a computer-readable medium that includes data strings representing language data, wherein each character of each data string is a character selected from the group consisting of standard ASCII, extended ASCII, and double byte characters;

an extended ASCII font data area stored on a computer-readable medium for storing font data related to extended ASCII characters that are not displayable using extended ASCII character font data stored in said font module; and

a double byte character font data area stored on a computer-readable medium for storing font data related to at least one double byte character.

6. A system according to Claim 5, wherein characters in a data string that are extended ASCII characters and standard ASCII characters, if any, having ASCII codes less than a selected escape code are stored by their ASCII representations in said string data area, while extended ASCII characters and standard ASCII, if any, having ASCII codes at least as great as the selected escape code and ASCII characters that identify the start of 16 bit double byte characters are encoded into 16 bit values and the encoded values are stored in said string data area.

7. A system according to Claim 5, wherein standard ASCII characters and extended ASCII characters having ASCII codes less than the selected escape code are

-41-

stored by their ASCII code representations in the string data area, and wherein extended ASCII characters having ASCII codes at least as great as the selected escape code and ASCII characters that identify the start of 16 bit double byte characters are encoded into 16 bit values and the encoded values are stored in the string data area.

5   8. A system according to Claim 7, wherein double byte characters are sequentially encoded and the 16 bit encoded values representing the double byte characters are stored in said string data area, wherein font data associated with the double byte characters is stored in the double byte character font data area, and wherein at least one extended ASCII character is encoded in said string data area with an escape code

10   preceding the ASCII representation of the extended ASCII character, and wherein if the extended ASCII character is not displayable with extended ASCII character font data stored in said font module, data for the extended ASCII character is stored in said extended ASCII font data area.

       9. A system according to Claim 8, wherein double byte characters are

15   sequentially encoded such that the first double byte character is represented by a two-byte code having a first byte that is one value greater than the escape code and a second byte equal to zero, and wherein remaining unique double byte characters are encoded with sequential 16 bit code values.

       10. A system according to Claim 9, wherein the escape code is selected as E0

20   hexadecimal, wherein double byte characters are sequentially encoded in said string data area such that the first double byte character is encoded as E100 hexadecimal and the remaining unique double byte characters are encoded with sequential 16 bit codes from E101 to FFFF hexadecimal, and wherein the extended ASCII characters having ASCII codes at least as great as the selected escape code are encoded in the string data area as a

25   16 bit code with the selected escape code as the first byte and the code for the extended ASCII character as the second byte.

       11. A system according to Claim 10, wherein to display a data string, said display management module receives the token associated with the data string and a command to display the data string from the computer program and accesses the location in the string

AttyDktNo 42626/204667

data area where the data string is located and sequentially displays characters of the data string on the display terminal.

12. A system according to Claim 10, wherein if a character in the data string is less than 80 hexadecimal, then the character is a standard ASCII character and said display management module retrieves the ASCII character code stored in the string data area and displays the ASCII character on the display terminal using font data from said font data module.

13. A system according to Claim 10, wherein if a character in the data string is at least as great as 80 hexadecimal but less than the escape code, then the character is an extended ASCII character having an ASCII code less than the escape code, wherein said display management module retrieves the ASCII character code stored in the string data area, wherein if the ASCII character is displayable with font data stored in the font data module, said display management module displays the ASCII character on the display terminal using font data from said font data module, and wherein if the ASCII character is not displayable with font data stored in the font data module, said display management module displays the ASCII character on the display terminal using font data from said extended ASCII font data area.

14. A system according to Claim 10, wherein if a character in the data string is equal to E0 hexadecimal, then the next character in the data string is an extended ASCII character and said display management module retrieves the character code stored in the next byte of the string data area, and wherein said display management module uses the character code to display the extended ASCII character on the display terminal.

15. A system according to Claim 10, wherein if a character in the data string is greater than the escape code, then the character combined with the next consecutive character in the data string represents a 16 bit double byte character.

16. A system according to Claim 15, wherein the escape code is E0 hexadecimal and the first encoded double byte character is E100 hexadecimal, wherein to display a double byte character said display management module subtracts E100 hexadecimal from

-43-

the character and multiplies the character by a size value representing the pixel display size of a double byte character, wherein the calculated value represents the location offset of the double byte character font data for the character stored in the double byte character font data area, and wherein said display management module uses the calculated value to

5   locate and use the double byte character font data stored in the double byte character font area to display the double byte character on the display terminal.

17. A system according to Claim 4, wherein said extended ASCII font data area includes font data for extended ASCII characters not stored in said font data module, wherein when said display management module receives a command to display a

10   character having associated font data stored in said font module, said display management module locates the font data associated with the character in said font module and using the font data displays the character on the display terminal, and wherein when said display management module receives a command to display an extended ASCII character not having associated font data stored in said font module, said

15   display management module locates the font data associated with the character in said extended ASCII font data area and using the font data displays the character on the display terminal.

18. A system according to Claim 1 further comprising a plurality of font modules each containing font data for displaying characters in a different font, and wherein when

20   said display management module receives a command to display a character in a preselected font, said display management module accesses the font data module associated with the preselected font, locates the font data associated with the character code in said font module and using the font data displays the character on the display terminal.

25   19. A system according to Claim 1 further comprising a logo module stored in a computer-readable medium and associated with said display management module, wherein said logo module contains graphical data for display of at least one logo, and wherein when said display management module receives a command to display the logo,

-44-   AttyDktNo 42626/204667

said display management module locates the graphical data associated with the logo in said logo module and using the logo data displays the logo on the display terminal.

20. A system according to Claim 19, wherein said logo module contains graphical data for displaying at least two logos, and wherein said display management module simultaneously displays graphical data on the display terminal for all logos.

21. A system according to Claim 1 wherein said display management module displays a progress bar on the display terminal.

22. A system according to Claim 1, wherein said display management module displays a box on the display terminal defining an area on the display terminal.

23. A system according to Claim 22, said display management module defines an area on the display terminal, and wherein as said display management module displays data within the defined area of the display terminal, said display management module scrolls the data displayed within the defined area, while any data displayed on other portions of the display terminal remain at the same position.

24. A system according to Claim 1, wherein display management module determines whether the display terminal operates in a graphic mode, and wherein if the display terminal does not operate in a graphic mode, said display management module operates in a text mode and displays only text characters on the display terminal.

25. A method for managing the display of information output by a computer program on a display terminal comprising:

providing on a computer-readable medium data modules containing text and graphical data used by the computer program to display information on the display terminal;

receiving a command from the computer program to display data on the display terminal;

retrieving data associated with the command from the data modules; and

displaying the data associated with the command on the display terminal, wherein storage of the text and graphical data in the data modules eliminates the requirement that

-45-

the text and graphical data for displaying information on the display terminal be contained in the computer program.

26. A method according to Claim 25, wherein said providing step provides at least one data module that is a language data module including data strings representing language data, wherein each data string is stored in the language data module and designated by a token, and wherein to display a data string, said receiving step receives a token associated with the data string and a command to display the data string from the computer program and based on the token said retrieving step accesses the language data module and retrieves the data string associated with the token, and wherein said displaying step displays the data string on the display terminal.

27. A method according to Claim 26, wherein said providing step provides a plurality of language data modules on a computer-readable medium, wherein each language data module includes data strings representing language data in a selected language, and wherein said providing step further provides a main language module header stored on a computer-readable medium comprising individual pointers indicating the location in a computer-readable storage medium in which each language data module is located, wherein to display a data string in a selected language, said receiving step receives a token and a command to display the data string from the computer program, wherein said retrieving step accesses the main language module header and retrieves the pointer associated with the language data module corresponding to a preselected desired language for displaying the data string, wherein said retrieving step, using the pointer, accesses the language data module and retrieves the data string associated with the token, and wherein said displaying step displays the data string on the display terminal in the desired language.

28. A method according to Claim 26, wherein said providing step further provides a font module stored in a computer-readable medium, wherein the font module contains font data for displaying 256 standard and extended ASCII characters.

AttyDktNo 42626/204667

29. A method according to Claim 28, wherein said providing step provides at least one data module that is a language data module, wherein said providing step provides:

a string data area stored on a computer-readable medium that includes data strings representing language data, wherein each character of each data string is a character selected from the group consisting of standard ASCII, extended ASCII, and double byte characters;

an extended ASCII font data area stored on a computer-readable medium for storing font data related to extended ASCII characters that are not displayable using the extended ASCII character font data stored in the font module ; and

a double byte character font data area stored on a computer-readable medium for storing font data related to at least one double byte character.

30. A method according to Claim 29, wherein said providing step provides a string data area wherein characters in a data string that are extended ASCII characters and standard ASCII characters, if any, having ASCII codes less than a selected escape code are stored by their ASCII representations in the string data area, while extended ASCII characters and standard ASCII, if any, having ASCII codes at least as great as the selected escape code and ASCII characters that identify the start of 16 bit double byte characters are encoded into 16 bit values and the encoded values are stored in the string data area.

31. A method according to Claim 29, wherein said providing step provides a string data area wherein characters in a data string that are standard ASCII characters and extended ASCII characters having ASCII codes less than a selected escape code are stored by their ASCII representations in the string data area, while extended ASCII characters having ASCII codes at least as great as the selected escape code and ASCII characters that identify the start of 16 bit double byte characters are encoded into 16 bit values and the encoded values are stored in the string data area.

32. A method according to Claim 29, wherein said providing step provides a string data area including double byte characters that are sequentially encoded and the

-47-   AttyDktNo 42626/204667

encoded values representing the double byte characters are stored in the string data area, wherein font data associated with the double byte characters is stored in the double byte character font data area, and wherein at least one extended ASCII character is encoded as a 16 bit value in the string data area with an escape code preceding the ASCII

5   representation of the extended ASCII character, and wherein if the extended ASCII character is not displayable with the extended ASCII character font data stored in said font module, data for the extended ASCII character is stored in the extended ASCII font data area.

33. A method according to Claim 32, wherein said providing step provides a

10   string data area including double byte characters that are sequentially encoded such that the first double byte character is represented by a two-byte code having a first byte that is one value greater than the escape code and a second byte equal to zero, and wherein remaining unique double byte characters are encoded with sequential 16 bit code values.

34. A method according to Claim 33, wherein the escape code is selected as E0

15   hexadecimal, wherein said providing step provides a string data area including double byte characters that are sequentially encoded in the string data area such that the first double byte character is encoded as E100 hexadecimal and the remaining unique double byte characters are encoded with sequential 16 bit values from E101 to FFFF hexadecimal, and wherein the extended ASCII characters having ASCII codes at least as

20   great the selected escape code are encoded in the string data area as a 16 bit code with the selected escape code as the first byte and the code for the extended ASCII character as the second byte.

35. A method according to Claim 34, wherein to display a data string, said receiving step receives the token associated with the data string and a command to

25   display the data string from the computer program, wherein said retrieving step accesses the location in the string data area where the data string is located, and wherein said display step sequentially displays characters of the data string on the display terminal.

36. A method according to Claim 35, wherein if a character in the data string is less than 80 hexadecimal, then the character is a standard ASCII character and said

-48-   AttyDktNo 42626/204667

retrieving step retrieves the ASCII character code stored in the string data area and said displaying step displays the ASCII character on the display terminal using font data from the font data module.

37. A method according to Claim 35, wherein if a character in the data string is at least as great as 80 hexadecimal but less than the escape code, then the character is an extended ASCII character having an ASCII code less than the escape code, wherein said retrieving step retrieves the extended ASCII character code stored in the string data area, wherein if the ASCII character is displayable with font data stored in the font data module, said displaying step displays the ASCII character on the display terminal using font data from said font data module, and wherein if the ASCII character is not displayable with font data stored in the font data module, said displaying step displays the ASCII character on the display terminal using font data from the extended ASCII font data area in the language module.

38. A method according to Claim 35, wherein if a character in the data string is equal to the selected escape code of E0 hexadecimal, then the next character in the data string is an extended ASCII character, wherein said retrieving step retrieves the character code stored in the next byte of the string data area, and wherein said displaying step uses the character code to display the extended ASCII character on the display terminal.

39. A method according to Claim 35, wherein if a character in the data string is greater than the escape code, then the character combined with the next consecutive character in the data string represents a 16 bit double byte character.

40. A method according to Claim 39, wherein the escape code is E0 hexadecimal and the first encoded double byte character is E100 hexadecimal, wherein to display a double byte character said retrieving step subtracts E100 hexadecimal from the character and multiplies the character by a size value representing the pixel display size of a double byte character, wherein the calculated value represents the location offset of the double byte character font data for the character stored in the double byte character font data area, and wherein said retrieving step uses the calculated value to locate the double byte

-49-

AttyDktNo 42626/204667

character font data stored in the double byte character font area, and wherein said displaying step displays the double byte character on the display terminal.

41. A method according to Claim 25, wherein said providing step further provides a font module stored in a computer-readable medium, wherein the font module contains font data for displaying 256 standard and extended ASCII characters, wherein the extended ASCII font data area includes font data for extended ASCII characters not stored in the font data module, wherein to display a character having associated font data stored in the font module, said retrieving step locates the font data associated with the character in the font module and said displaying step uses the font data to display the ASCII character on the display terminal, and wherein to display a character not having associated font data stored in the font module, said retrieving step locates the font data associated with the character in the extended ASCII font data area and said displaying step uses the font data to display the ASCII character on the display terminal .

42. A method according to Claim 25, wherein said providing step provides a plurality of font modules each containing font data for displaying characters in a different font, and wherein when said receiving step receives a command to display a character in a preselected font, said retrieving step accesses the font data module associated with the preselected font and locates the font data associated with the character code in the font module, and wherein said displaying step, using the font data, displays the character on the display terminal.

43. A method according to Claim 25, wherein said providing step provides a logo module stored in a computer-readable medium, wherein the logo module contains graphical data for display of at least one logo, and wherein when said receiving step receives a command to display the logo, said retrieving step locates the graphical data associated with the logo in the logo module, and said displaying step, using the logo data, displays the logo on the display terminal.

44. A method according to Claim 43, wherein said providing step provides a logo module containing graphical data for displaying at least two logos, and wherein said

AttyDktNo 42626/204667

displaying step simultaneously displays graphical data on the display terminal for all logos.

45. A method according to Claim 25, wherein said displaying step displays a progress bar on the display terminal.

5   46. A method according to Claim 25, wherein said displaying step displays a text box on the display terminal defining an area on the display terminal.

47. A method according to Claim 46, wherein said displaying step defines an area on the display terminal, and wherein as said displaying step displays data within the defined area and scrolls the data displayed within the defined area, while any data

10   displayed on other portions of the display terminal remain at the same position.

48. A method according to Claim 25 further comprising the step of determining whether the display terminal operates in a graphic mode, and wherein if the display terminal does not operate in a graphic mode, said displaying step operates in a text mode and displays only text characters on the display terminal.

15   49. A computer program product for managing the display of information output by a computer program on a display terminal, wherein the computer program product comprises:

a computer-readable storage medium having computer readable program code means embodied in said medium, said computer-readable program code means

20   comprising:

first computer instruction means for providing on a computer-readable medium data modules containing text and graphical data used by the computer program to display information on the display terminal;

second computer instruction means for receiving a command from the

25   computer program to display data on the display terminal;

third computer instruction means for retrieving data associated with the command from the data modules; and

AttyDktNo 42626/204667

fourth computer instruction means for displaying the data associated with the command on the display terminal, wherein storage of the text and graphical data in the data modules eliminates the requirement that the text and graphical data for displaying information on the display terminal be contained in the computer program.

5   50. A computer program product according to Claim 49, wherein said first computer instruction means provides at least one data module that is a language data module including data strings representing language data, wherein each data string is stored in the language data module and designated by a token, and wherein to display a data string, said second computer instruction means receives a token associated with the

10   data string and a command to display the data string from the computer program and based on the token said third computer instruction means accesses the language data module and retrieves the data string associated with the token, and wherein said fourth computer instruction means displays the data string on the display terminal.

       51. A computer program product according to Claim 49, wherein said first

15   computer instruction means provides a plurality of language data modules on a computer-readable medium, wherein each language data module includes data strings representing language data in a selected language, and wherein said first computer instruction means further provides a main language module header stored on a computer-readable medium comprising individual pointers indicating the location in a computer-readable storage

20   medium in which each language data module is located, wherein to display a data string in a selected language, said second computer instruction means receives a token from the computer program and a command to display the data string, wherein said third computer instruction means accesses the main language module header and retrieves the pointer associated with the language data module corresponding to a preselected desired

25   language for displaying the data string, wherein said third computer instruction means, using the pointer, accesses the language data module and retrieves the data string associated with the token, and wherein said fourth computer instruction means displays the data string on the display terminal.

AttyDktNo 42626/204667

52. A computer program product according to Claim 50, wherein said first computer instruction means further provides a font module stored in a computer-readable medium, wherein the font module contains font data for displaying 256 standard and extended ASCII characters.

53. A computer program product according to Claim 52, wherein said first computer instruction means provides at least one data module that is a language data module, wherein said first computer instruction means provides:

a string data area stored on a computer-readable medium that includes data strings representing language data, wherein each character of each data string is a character selected from the group consisting of standard ASCII, extended ASCII, and double byte characters;

an extended ASCII font data area stored on a computer-readable medium for storing font data related to extended ASCII characters that are not displayable using the extended ASCII character font data stored in said font module; and

a double byte character font data area stored on a computer-readable medium for storing font data related to at least one double byte character.

54. A computer program product according to Claim 53, wherein said first computer instruction means provides a string data area wherein characters in a data string that are extended ASCII characters and standard ASCII characters, if any, having ASCII codes less than a selected escape code are stored by their ASCII representations in the string data area, while extended ASCII characters and standard ASCII, if any, having ASCII codes at least as great as the selected escape code and ASCII characters that identify the start of 16 bit double byte characters are encoded into 16 bit values and the encoded values are stored in the string data area.

55. A computer program product according to Claim 53, wherein said first computer instruction means provides a string data area wherein characters in a data string that are standard ASCII characters and extended ASCII characters having ASCII codes less than a selected escape code are stored by their ASCII code representations in the string data area, while extended ASCII characters having ASCII codes at least as great as

AttyDktNo 42626/204667

the selected escape code and ASCII characters that identify the start of 16 bit double byte characters are encoded into 16 bit values and the encoded values are stored in the string data area.

56. A computer program product according to Claim 53, wherein said first computer instruction means provides a string data area including double byte characters that are sequentially encoded and the encoded values representing the double byte characters are stored in the string data area, wherein font data associated with the double byte characters is stored in the double byte character font data area, and wherein at least one extended ASCII character is encoded as a 16 bit value in the string data area with an escape code preceding the ASCII representation of the extended ASCII character, and wherein if the extended ASCII character is not displayable with the extended ASCII character font data stored in said font module, data for the extended ASCII character is stored in the extended ASCII font data area.

57. A computer program product according to Claim 55, wherein said first computer instruction means provides a string data area including double byte characters that are sequentially encoded such that the first double byte character is represented by a two-byte code having a first byte that is one value greater than the escape code and a second byte equal to zero, and wherein the remaining unique double byte characters are encoded with sequential 16 bit code values.

58. A computer program product according to Claim 57, wherein the escape code is selected as E0 hexadecimal, wherein said first computer instruction means providing step provides a string data area including double byte characters that are sequentially encoded in the string data area such that the first double byte character is encoded as E100 hexadecimal and the remaining unique double byte characters are encoded with sequential 16 bit values from E101 to FFFF hexadecimal, and wherein the extended ASCII characters having ASCII codes at least as great as the selected escape code are encoded in the string data area as a 16 bit code with the selected escape code as the first byte and the code for the extended ASCII character as the second byte.

AttyDktNo 42626/204667

59. A computer program product according to Claim 58, wherein to display a data string, said second computer instruction means receives the token associated with the data string and a command to display the data string from the computer program, wherein said third computer instruction means accesses the location in the string data area where the data string is located, and wherein said fourth computer instruction means sequentially displays characters of the data string on the display terminal.

60. A computer program product according to Claim 59, wherein if a character in the data string is less than 80 hexadecimal, then the character is a standard ASCII character and said third computer instruction means retrieves the ASCII character code stored in the string data area and said fourth computer instruction means displays the ASCII character on the display terminal using font data from the font data module.

61. A computer program product according to Claim 59, wherein if a character in the data string is at least as great as 80 hexadecimal but less than the escape code, then the character is an extended ASCII character having an ASCII code less than the escape code, wherein said third computer instruction means retrieves the extended ASCII character code stored in the string data area, wherein if the ASCII character is displayable with font data stored in the font data module, said fourth computer instruction means displays the ASCII character on the display terminal using font data from said font data module, and wherein if the ASCII character is not displayable with font data stored in the font data module, said fourth computer instruction means displays the ASCII character on the display terminal using font data from the extended ASCII font data area in the language module.

62. A computer program product according to Claim 59, wherein if a character in the data string is equal to the selected escape code of E0 hexadecimal, then the next character in the data string is an extended ASCII character, wherein said third computer instruction means retrieves the character code stored in the next byte of the string data area, and wherein said fourth computer instruction means uses the character code to display the extended ASCII character on the display terminal.

AttyDktNo 42626/204667

63. A computer program product according to Claim 57, wherein if a character in the data string is greater than the escape code, then the character and a next consecutive character in the data string together represent a 16 bit double byte character.

64. A computer program product according to Claim 63, wherein the escape code is E0 hexadecimal and the first encoded double byte character is E100 hexadecimal, wherein said third computer instruction means subtracts E100 hexadecimal from the character and multiplies the character by a size value representing the pixel display size of a double byte character, wherein the calculated value represents the location offset of the double byte character font data for the character stored in the double byte character font data area, and wherein said third computer instruction means uses the calculated value to locate the double byte character font data stored in the double byte character font area, and wherein said fourth computer instruction means displays the double byte character on the display terminal.

65. A computer program product according to Claim 50, wherein said first computer instruction means further provides a font module stored in a computer-readable medium, wherein the font module contains font data for displaying 256 standard and extended ASCII characters, wherein the extended ASCII font data area includes font data for extended for extended ASCII characters not stored in the font data module, wherein to display a character having associated font data in the font module, said third computer instruction means locates the font data associated with the ASCII character in the font module and said fourth computer instruction means uses the font data to display the ASCII character on the display terminal, and wherein to display a character not having associated font data stored in the font module, said third computer instruction means locates the font data associated with the character in the extended ASCII font data area and said displaying step uses the font data to display the ASCII character on the display terminal.

66. A computer program product according to Claim 50, wherein said first computer instruction means provides a plurality of font modules each containing font data for displaying characters in a different font, and wherein when said second computer

AttyDktNo 42626/204667

instruction means receives a command to display a character in a preselected font, said third computer instruction means accesses the font data module associated with the preselected font and locates the font data associated with the character code in the font module, and wherein said fourth computer instruction means, using the font data,

5 displays the character on the display terminal.

67. A computer program product according to Claim 50, wherein said first computer instruction means provides a logo module stored in a computer-readable medium, wherein the logo module contains graphical data for display of at least one logo, and wherein when said second computer instruction means receives a command to

10 display the logo, said retrieving step locates the graphical data associated with the logo in the logo module, and said fourth computer instruction means, using the logo data, displays the logo on the display terminal.

68. A computer program product according to Claim 67, wherein said first computer instruction means provides a logo module containing graphical data for

15 displaying at least two logos, and wherein said fourth computer instruction means simultaneously displays graphical data on the display terminal for all logos.

69. A computer program product according to Claim 50, wherein said fourth computer instruction means displays a progress bar on the display terminal.

70. A computer program product according to Claim 50, wherein said fourth

20 computer instruction means displays a text box on the display terminal defining an area on the display terminal.

71. A computer program product according to Claim 70, wherein said fourth computer instruction means defines an area on the display terminal, and wherein as said fourth computer instruction means displays data within the defined area and scrolls the

25 data displayed within the defined area, while any data displayed on other portions of the display terminal remain at the same position.

72. A computer program product according to Claim 50 further comprising fifth computer instruction means for determining whether the display terminal operates in a

-57-

graphic mode, and wherein if the display terminal does not operate in a graphic mode, said fourth computer instruction means operates in a text mode and displays only text characters on the display terminal.